Developing a Distributed Data Dictionary Service

Jim U'Ren, Dan Crichton & Steve Hughes

June 20, 2001

Abstract

In an interconnected world, applications need to interconnect in much the same way that computers are connected today: i.e. through the use of well established standards-based technologies. Data dictionaries are the keys to these applications outlining the meaning and structure of the information contained within them. What is currently missing is an architecture and mechanisms that allows data dictionaries to be systematically linked to each other thus enabling connectivity between applications. This paper will explore the use of combining the Lightweight Directory Access Protocol (LDAP) and the ISO/IEC 11179 Data Element Set as mechanisms for standardizing the structure and communication links between data dictionaries.

Background

In the last decade, with the widespread adoption of the Internet, we have witnessed tremendous strides in the ability to access information worldwide. With the click of a button, we can read electronic editions of newspapers, check account balances, trade stocks, access weather reports, listen to music, view still and motion video, browse library catalogues and shop for virtually all goods or services. All of this is the result of standards that allow virtually any device to communicate on the Internet providing they adhere to some basic rules and protocols. This convergence on a basic set of networking standards provides a basic capability that allows for ongoing enhancements and development. The result is an open networking environment that supports on going evolutionary development allowing a broad range of users and developers to participate.

But while we can access and view information, reusing and integrating information is still problematic. For example, cost information is usually available on-line in most organizations and information about the same organization’s products or services is also available on-line but a common problem occurs when cost information must be integrated with planning or product information. Considerable human interpretation and manual re-entry of data is usually required. The issues becomes even more complex when organizations merge or partner and need to integrate information into the new organizational structures. This is largely due to the fact that the information about these datasets, better known as data dictionaries, are not readily available in easy to use formats. Data dictionaries are repositories of information about a datasets that provide a way of managing the semantics and data elements that make up an application. [1]

[1] IBM Dictionary of Computing, p. 168

Data dictionaries are currently published in a number of ways:

- As an electronic file (PDF file, MS Word, MS Access, Excel Spreadsheet, etc) – where all elements in the dictionary are morphed into a blob. Is human readable but not very machine interpretable.
- As an on-line database (HTML web page, SQL DB, Web-based CGI access to database) – where each entry is a record in the database; automated routines such as validations are possible.
- In hardcopy printed format – human readable but not machine accessible

Despite, the fact that data dictionaries are pushed in a variety of formats, it should be noted that there are many examples of very successful data dictionary implementations that address issues important to the community that uses the application and the data it manages. Current domain specific implementations solve problems such as:

- Validating the integrity of newly entered data
- Providing vocabulary used in the applications
- Providing a database of data elements that allows reuse within the application domain

Despite these successful domain specific data dictionary implementations, there are a number of fundamental problems that need to be addressed to take data dictionaries to the next level.

The Problem

No piece of information is an island unto itself. Every piece of information has relationships with one or more pieces of information and the ability to easily access and integrate data dictionary information is essential if these relationships are to be understood.

The fundamental problem that must be addressed is that data dictionaries are not available on-line using consistent structures and access mechanisms. Locating, accessing and reusing data dictionary information requires considerable manual effort and this creates a barrier to data integration and reuse. Most data dictionaries are currently structured and published in a manner that serves the specific purposes of a local target community or domain; they are published in a manner that does not provide easy access or use by groups or applications outside the domain. This has resulted in limited reuse and integration of information between domains and applications.

Another problem that must be addressed is clarifying terms used by the data dictionary community. The concept of data dictionary means different things to different people. To some it is a vocabulary list of terms used in an application. To others it is a collection of the descriptions of each field of a dataset. To others, it is the relationship of a database field to each other and the relationships to fields in other datasets.

There is a general need for a service that will help users and applications quickly learn the similarities and differences between data sets. Integrating and reusing information today that is distributed across different databases is similar to sharing information between computers prior to the Internet. i.e. prior to adoption of Internet standards, networking applications often consisted of a “rats-maze” of cables and networking protocols. Today we have equally confusing and discontinuous datasets across the Net because even though they may be on-line they require separate access mechanisms and are in incompatible formats.

A Solution

A proposed solution is to use two established standards to provide common access and common data structures: the Internet standard, LDAP (Lightweight Directory Access Protocol), to provide the a common access mechanism and the ISO/IEC 11179 data element set to provide a common data structure. This approach allows for an open scalable, extensible and cost effective way of publishing, accessing and integrating data dictionaries.

LDAP (Lightweight Directory Access Protocol) is a directory service developed at the University of Michigan and standardized through the Internet standards process.[2] LDAP is both an information model and a protocol for querying and manipulating directories. LDAP's overall data and namespace model is essentially that of X.500 but is designed to run directly over the TCP/IP stack.

In this context, a directory is like a database, but tends to contain more descriptive, attribute-value-based information. The information in a directory is generally read much more often than it is written. As a result, directories don't usually implement the more complicated transaction or roll-back schemes common to SQL databases that are best suited for high-volume complex updates. Directory updates are typically simple all-or-nothing changes, if they are allowed at all. Directories are tuned to give quick-response to high-volume lookup or search operations. They may have the ability to replicate information widely in order to increase availability and reliability, while reducing response time. When directory information is replicated, temporary inconsistencies between the replicas may be OK, as long as they get in sync eventually.

A central feature of LDAP is that it defines a global directory structure. LDAP essentially describes a directory web in much the same way that http and html are used to define and implement the global hypertext Web that we all use today. Anyone with an LDAP client may browse the global directory just as they can use a web browser to peruse the global Web. Additionally, with the help of Web-LDAP gateways and the LDAP URL specification, a web client can be used to browse both spaces.

LDAP identifies each entry in a database through the use of a mechanism known as *distinguished names* (DNs) that are organized in a hierarchy, each consisting of the name of an entry plus a path of names tracing the entry back to the root of the tree.

[2] Internet Standards Process, RFC 1310

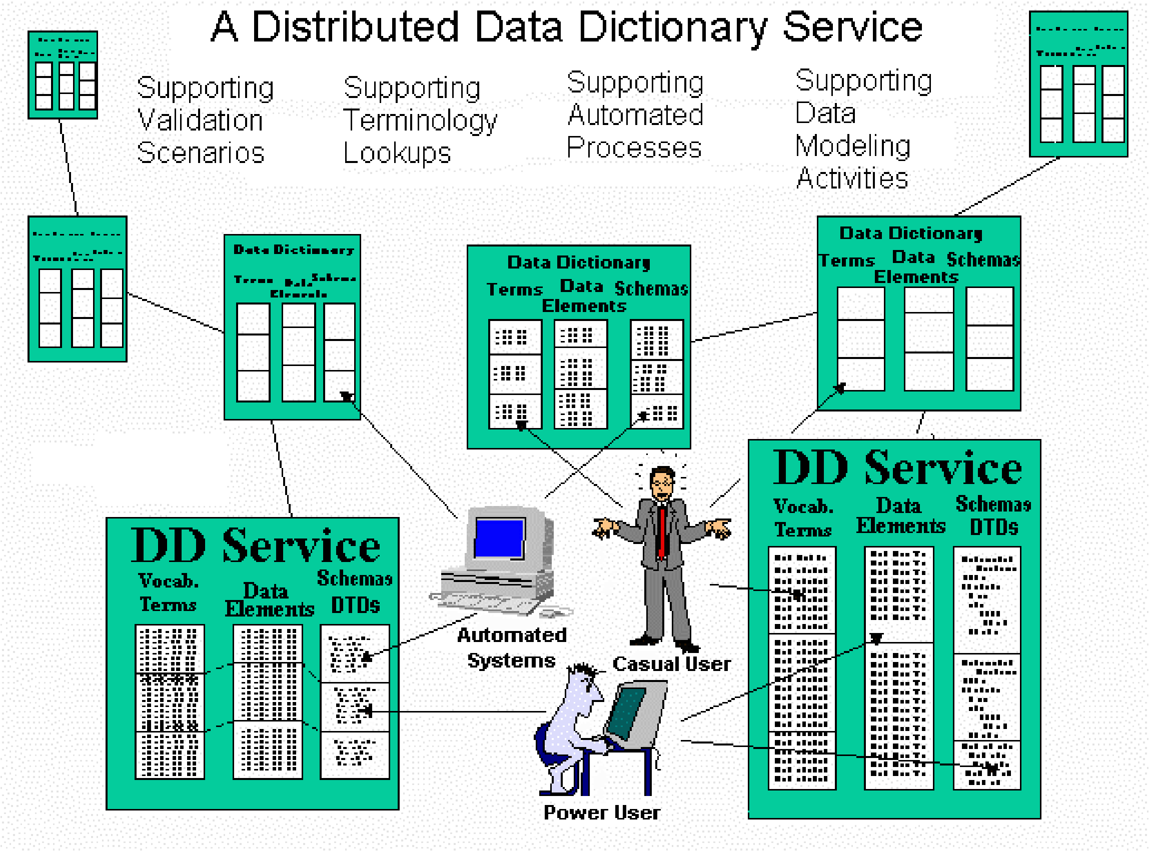

A Distributed Data Dictionary Service
Supporting Validation Scenarios
Supporting Terminology Lookups
Supporting Automated Processes
Supporting Data Modeling Activities
Data Dictionary
Terms
Data Elements
Schemas
DD Service
Vocab. Terms
Data Elements
Schemas DTDs
Automated Systems
Casual User
Power User

Figure 1

For example, ISO, as an international standards organization initiates and manages numerous standards activities and as such would be the top entry in the hierarchy for ISO standards activities with a DN of `“dc=ISO”.` A base DN defines the top of the namespace that the server is responsible for, analogous to a DNS zone on the Internet. Within ISO, there are hundreds of activities, such as STEP, more formally known as ISO 10303, or MPEG, known as ISO/IEC 14496. The next level down the ISO tree would contain a DN `“dc=10303,dc=ISO”` for STEP and `“dc=14496, dc=ISO”` for MPEG. Further, breakdowns may be required as is the case for STEP in which there are numerous parts contained with ISO 30303. One such part is AP233, STEP’s Systems Engineering initiative, more formally know as Part233. AP233 represents a domain or namespace in which vocabulary, data elements and data structures are associated in a common application environment. Following this DN naming logic, the term “behaviour” in the STEP System Engineering domain would have a DN such as:

```
"cn=behaviour, dc=vocabulary, dc=Part233, dc=10303, dc=ISO"
```

As activities occur within a domain, additional nodes are added within and further down the hierarchy. The DNs become long as is necessary to identify an application domain that has vocabulary, data elements and data structures that support it.. It is important to note that a long

DN is not an issue for end users, because client applications will be doing the searches and then displaying the attributes the user needs to see similar to long URLs and obscure IP addresses.

An important feature that LDAP provides to a distributed data dictionary service is a mechanism called a *referral* that links directories hierarchically. A referral provides a way for servers to refer client requests to additional directory servers. Referrals allow the distribution of namespace information among multiple servers, provide knowledge of where data resides within a set of interrelated servers and route requests to the appropriate server. Referrals are the mechanism that can help create a data dictionary web. This would work very similar to IP address resolution in which DNS servers are linked hierarchically referring requests to subordinate or superordinate servers in order to resolve the packet that is being sent. [3] In an LDAP-based data dictionary service, a user or application request for a vocabulary item or data element can be automatically referred to an authoritative server thus saving time and speeding up the process of obtaining relevant information about a dataset.

Hierarchical routing is not new; many systems have used it before. An example is the U.S. telephone system where a 10-digit phone number is divided into a 3-digit area code, 3-digit exchange, and 4 digit connection. The advantage of using a hierarchical address is that it accommodates large growth because it means a given gateway does not need to know as much detail about distant destinations as it does about local ones. One disadvantage is that choosing a hierarchical structure is difficult, and it often becomes difficult to change a hierarchy once it has been established. [4]

LDAP databases can be contained in a variety of formats:

- LDIF - used to represent LDAP entries in a simple text format.
- LDBM – a native LDAP back-end database format
- SQL database – e.g. Oracle, SQLserver, Informix, MySQL, PostGres, DB2, etc.
- Flat file database – e.g MS Access, FoxPro, etc.

A key feature of using LDAP as the communication protocol for the data dictionary service is the fact that the client-server architecture allows considerable flexibility in terms of how information is obtained from the service. End users or applications can access information from the service using a variety of exiting client software, APIs (application programming interfaces) and languages. The flexibility inherent with LDAP's client-server architecture means the interfaces to the data dictionary service can be custom tailored to the specific needs of the target community whether they be casual users needing to lookup terms and acronyms, process owners needing to update and maintain namespace information or automated processes needing to validate the integrity of a given dataset.

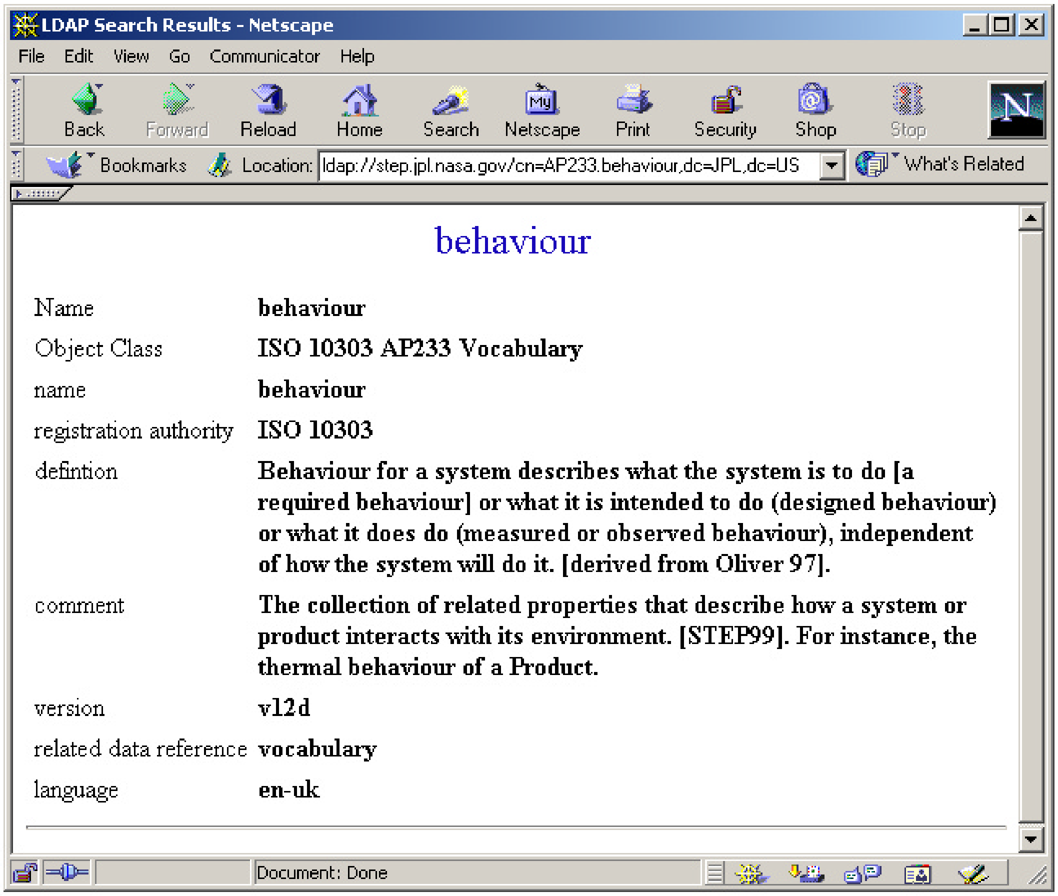
LDAP clients have been written in C/C++, Java, Perl, Python, PHP, VisualBasic, ColdFusion but perhaps the most significant and most common interface to LDAP services is through the LDAP

[3] T. Howes, p. 296

[4] Comer, p. 199.

URL[5] that allows any web browser supporting the LDAP URL standard to query LDAP services directly.

Below is an example of an LDAP query using the LDAP URL in Netscape 4.77.



ldap://step.jpl.nasa.gov/cn=AP233.behaviour,dc=JPL,dc=US

behaviour

Name	**behaviour**
Object Class	**ISO 10303 AP233 Vocabulary**
name	**behaviour**
registration authority	**ISO 10303**
defintion	**Behaviour for a system describes what the system is to do [a required behaviour] or what it is intended to do (designed behaviour) or what it does do (measured or observed behaviour), independent of how the system will do it. [derived from Oliver 97].**
comment	**The collection of related properties that describe how a system or product interacts with its environment. [STEP99]. For instance, the thermal behaviour of a Product.**
version	**v12d**
related data reference	**vocabulary**
language	**en-uk**

Figure 2

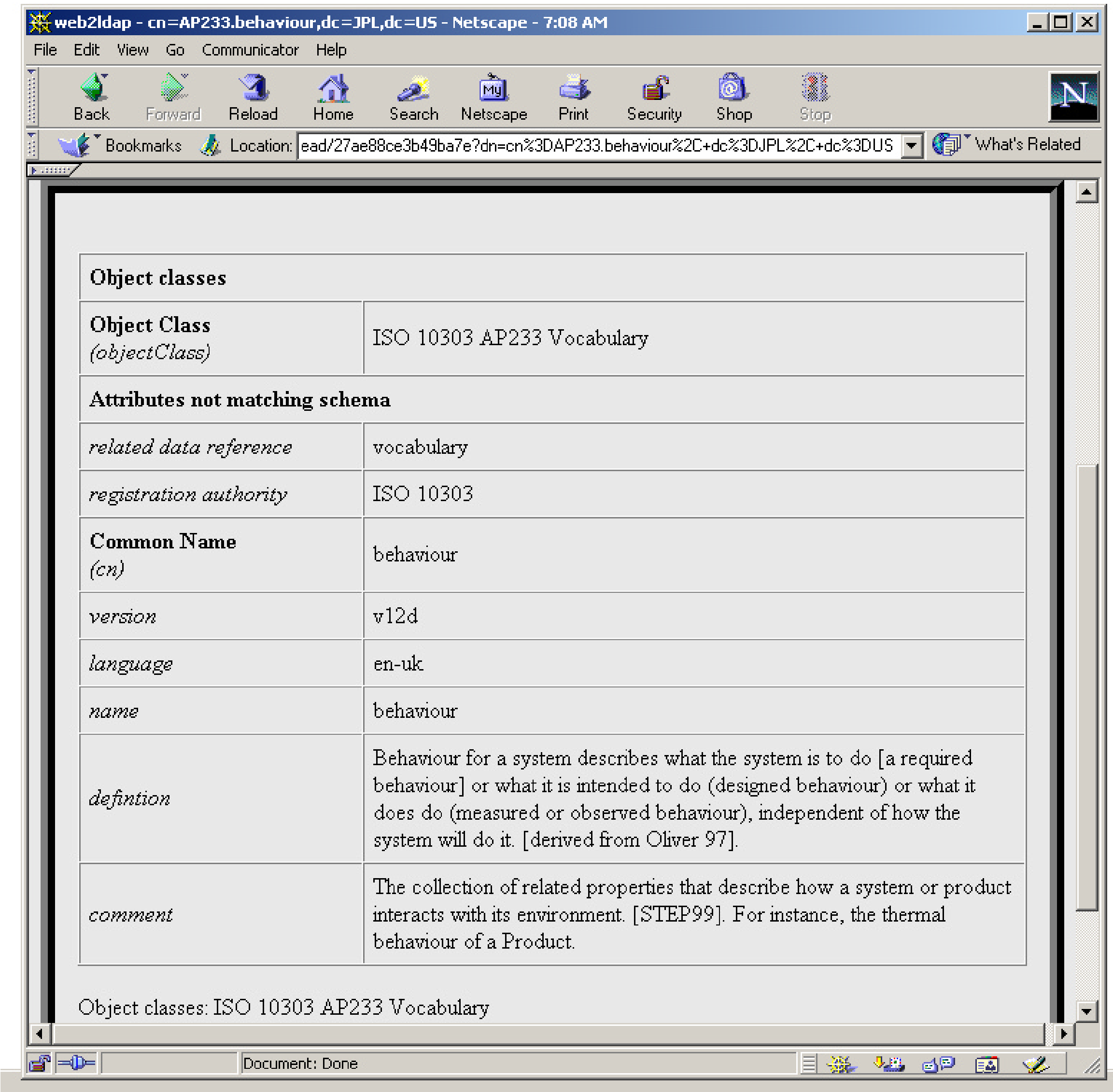
Note the LDAP URL in Netscape's Location Window that specifies, from left to right, LDAP as the protocol, the name of the server and the DN.

```
ldap://step.jpl.nasa.gov/cn=AP233.behaviour,dc=JPL,dc=US
|Protocol|  Name of Server |    DN  ( LDAP distinguished name)|
```

The LDAP URL is just one of serveral Web Interfaces to LDAP information. Using CGI and other existing Web technologies, a Web interface to a LDAP service can be easily added and customized.

[5] The LDAP URL Format, RFC 1959

Below is an example of an open source Python-based LDAP client called Web2LDAP, developed by Michael Stroeder. [6]


web2ldap - cn=AP233.behaviour,dc=JPL,dc=US - Netscape - 7:08 AM
File
Edit
View
Go
Communicator
Help
Back
Forward
Reload
Home
Search
Netscape
Print
Security
Shop
Stop
Bookmarks
Location: ead/27ae88ce3b49ba7e?dn=cn%3DAP233.behaviour%2C+dc%3DJPL%2C+dc%3DUS
What's Related
Document: Done

Object classes	
Object Class *(objectClass)*	ISO 10303 AP233 Vocabulary
Attributes not matching schema	
related data reference	vocabulary
registration authority	ISO 10303
Common Name *(cn)*	behaviour
version	v12d
language	en-uk
name	behaviour
defintion	Behaviour for a system describes what the system is to do [a required behaviour] or what it is intended to do (designed behaviour) or what it does do (measured or observed behaviour), independent of how the system will do it. [derived from Oliver 97].
comment	The collection of related properties that describe how a system or product interacts with its environment. [STEP99]. For instance, the thermal behaviour of a Product.

Object classes: ISO 10303 AP233 Vocabulary

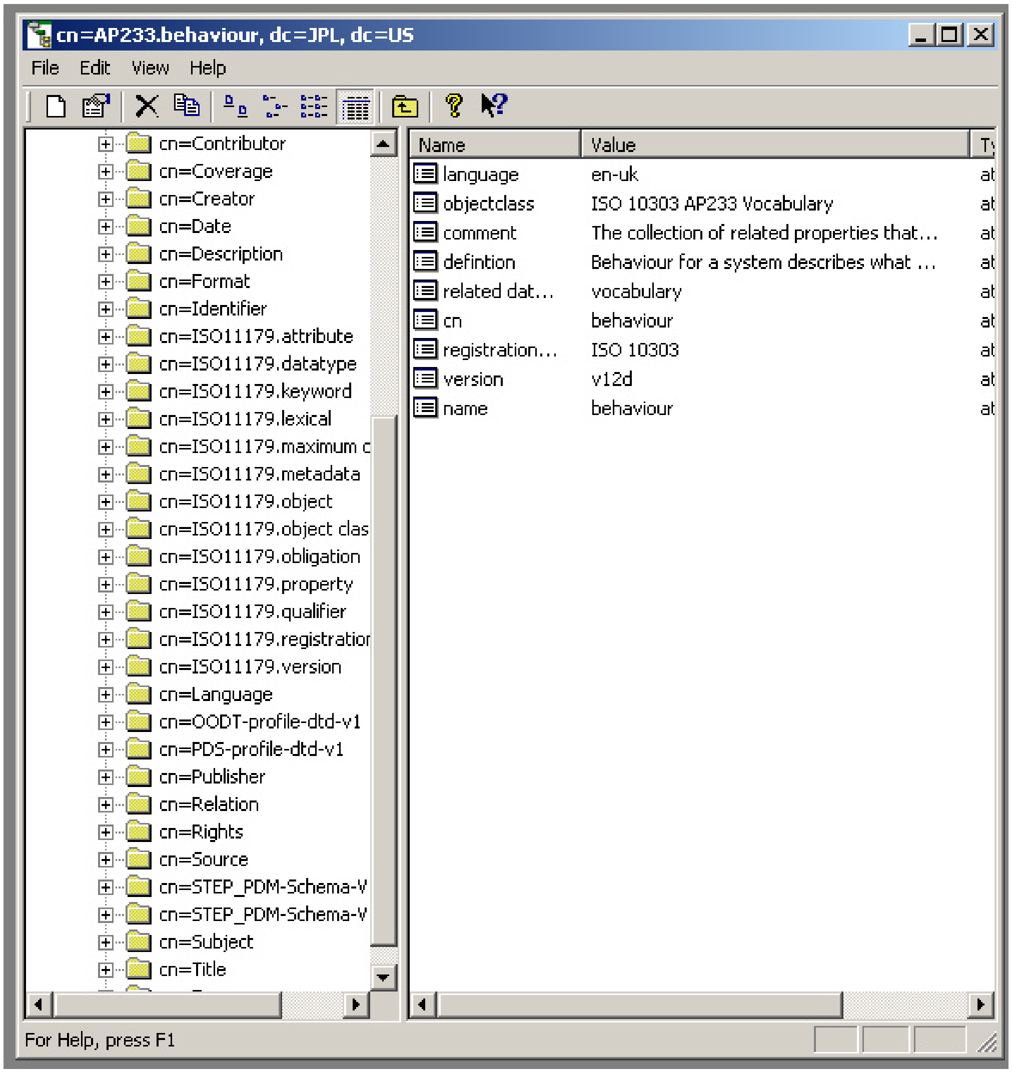
Figure 3

Examples of other open source LDAP software supporting Web interfaces are:

- LDAP Browser/Editor - a Java-based browser/editor, http://www.iit.edu/~gawojar/ldap
- GQ - a GTK based LDAP browser and editor, http://biot.com/gq/
- kldap - a KDE-based LDAP browser and editor, http://www.mountpoint.ch/oliver/kldap/
- Frood - a Gtk/PerLDAP based client., `http://frood.sourceforge.net/`

[6] web2ldap , http://www.web2ldap.de/ , Michael Ströder <michael@stroeder.com>

Below in Figure 4 is a screen from a “native client”, called LDAP Browser, a *freeware*[7] MS-Windows based client developed by Softera. [8]



cn=Contributor
cn=Coverage
cn=Creator
cn=Date
cn=Description
cn=Format
cn=Identifier
cn=ISO11179.attribute
cn=ISO11179.datatype
cn=ISO11179.keyword
cn=ISO11179.lexical
cn=ISO11179.maximum c
cn=ISO11179.metadata
cn=ISO11179.object
cn=ISO11179.object clas
cn=ISO11179.obligation
cn=ISO11179.property
cn=ISO11179.qualifier
cn=ISO11179.registratior
cn=ISO11179.version
cn=Language
cn=OODT-profile-dtd-v1
cn=PDS-profile-dtd-v1
cn=Publisher
cn=Relation
cn=Rights
cn=Source
cn=STEP_PDM-Schema-V
cn=STEP_PDM-Schema-V
cn=Subject
cn=Title

Name	Value	T
language	en-uk	at
objectclass	ISO 10303 AP233 Vocabulary	at
comment	The collection of related properties that…	at
defintion	Behaviour for a system describes what …	at
related dat…	vocabulary	at
cn	behaviour	at
registration…	ISO 10303	at
version	v12d	at
name	behaviour	at

Figure4

Note: Native clients often include platform specific interface features but are only available to those using the computing platform the application was written for.

[7] Freeware in this context means freely available for unconditional use; as opposed to shareware in which users are allowed to use software with certain limitations such as how long it can be used before purchasing a license or in what setting the software maybe used. Freeware in this context does not mean open source; only executable binaries are available for downloading.

[8] LDAP Browser by Softerra, http://www.ldapbrowser.com/

To compliment LDAP as a protocol standard for accessing data dictionary information, the ISO/IEC 11179 standard, formally known as the *Specification and Standardization of Data Elements*, provides structure and methodology for applying classifications to data elements. There are several purposes for applying classification to data elements. Classification assists users finding a single data element from among many data elements, facilitates data administration analysis of data elements and, through inheritance, conveys semantic content that is often only incompletely specified by other attributes, such as names and definitions.[9]

The classification schemes accommodated in ISO/IEC 11179 have utility for:

- deriving and formulating abstract and application data elements
- ensuring appropriate attribute and attribute-value inheritance
- deriving names from a controlled vocabulary
- disambiguating (clarifying exact content of information)
- recognizing superordinate, coordinate, and subordinate data element concepts
- recognizing relationships among data element concepts and data elements
- assisting in the development of modularly designed names and definitions.[10]

ISO 11179 components can be assembled according to rules defined by a naming convention into a set of standardized names identifying data elements in a structured procedure, resulting in names which mirror the discipline of the data elements. In the same way, rules and guidelines for well-formed definitions are described. A set of standardized attributes, some with sample values, prescribe the base set of information to be recorded about each data element. Finally, procedures for registration of standard data elements provide a facility for sharing the benefits of standardization among organizations.[11]

To address implementation issues and provide guidelines for developers, an ISO/IEC 11179 Metadata Registry Implementation Coalition has been organized to provide a forum for information exchange on the implementation of metadata registries based on the ISO/IEC-11179. The Consortium consists of members interested in addressing ISO/IEC-11179 reference implementations of metadata registries, influencing commercial vendors to support ISO/IEC-11179 in their tools, developing methods to support metadata exchange between metadata registries, sharing information and lessons learned on implementation approaches, being an advocate and clearinghouse for metadata registry issues, and developing partnerships to support data management across organizations.[12]

[9] ISO/IEC 11179-2 , p 3.

[10] Ibid, p. 3.

[11] J. Newton, Application of Metadata Standards, p.1

[12] ISO/IEC 11179 Metadata Registry Implementation Coalition, http://www.sdct.itl.nist.gov/~ftp/l8/other/coalition/Coalition.htm

Building a data dictionary service[13] using LDAP as a communication protocol and ISO 11179 as a data structure standard has the following benefits:

- Open – based on publicly available standard specifications that are vendor independent
- Distributed – LDAP service is designed for distributed implementations; servers can be link hierarchically
- Modular – domains for dictionaries can be delineated on servers; servers can range in size based on hardware availability and network bandwidth. If and when performance begins to drag on one server, entire domain databases can be moved to new locations as needed.
- Flexible – LDAP server schema can be configured as flexible as the environment requires; minimum number of attributes can be required with additional attribute value pairs added as necessary and the client-server architecture provides a range of client from thin HTML browser access using LDAP URL specification to open source Java, Perl, Python or php clients to full featured commercial LDAP clients.
- Extensible – LDAP schemas can be extended as necessary; extensions can be implemented on local servers without affecting the server's links to other servers; this is already proven in existing LDAP/X.500 implementations.

Benefits from a Distributed Data Dictionary Service

A clarification of semantics – one of the fundamental problems that a affects communication is the meaning of words in a given context. When information with known interfaces is generated in domains, the common words with different meanings will affect the ability to share and integrate information.

Flexible implementation – minimal requirements on end users. There are both open source and commercial products that support LDAP servers and clients. LDAP schemas support a range of flexibility that can provide localized optimizations while still maintaining overall connectivity and integration.

Adapt existing data dictionaries – LDAP has the ability to "wrap" existing data dictionaries. There is commercial and open source software currently available for converting and/or wrapping existing electronic data dictionaries. This would allow the integration of existing on-line data dictionaries in database formats such as Oracle SQL, MS Access, etc. to become part of a distributed LDAP-based data dictionary service.

Scalable security - established and proven LDAP and Internet security mechanisms can be employed in layers as required to secure information contained in an LDAP data dictionary server.

[13] Note: the term service in this context follows the Internet client-server model in which a client is a piece of software that runs on one computer and sends a request to a service that usually runs on a second computer. A dictionary service in this context is not a service bureau; it is an on-line capability that responds to client requests.

Supported Use Cases and Scenarios

The following are four Use Cases and associated Scenarios for each Use Case. Note: these Use Cases and Scenarios are meant to be representative not exhaustive.

Terminology Lookup Use Case Scenarios

- **Resolving Ambiguous Terminology** - an end user, needing to clarify use and meaning of a word used in a specific context, performs a multi-domain vocabulary lookup across multiple DD services looking for published vocabulary of referenced domain
- **Finding the Correct Acronym** - an end user, confronted with a number of new acronyms used in a presentation, accesses a local DD service to look up the acronyms based within probable domains, thereby eliminating the alternative meanings e.g., searching for STEP standards work versus the JPL STEP project
- **Enabling Improved Search Engine Performance** - as a search engine scans through a document, it discovers a keyword list and finds a “reserved word”; the document includes a URI reference to a domain-specific vocabulary list in a support DD service; the search engine uses this vocabulary to be certain it is indexing the keywords in the right context
- **Building Glossaries for Technical Papers** - an engineer or scientist writing a technical paper, needs to include a glossary of relevant terms in the paper; by performing a multi-service search, terms and definitions that relate to the topic of the paper are quickly found and inserted into the paper with the corresponding attributions

Validation Scenarios

- **Validating Units of Measure** - A system integrator receives an MCAD geometry model (e.g., STEP AP203 Part 21 file) of a component to be integrated into any assembly; automatically, a standard validation routine is performed against the schema located in a data dictionary service that checks for use of the units of measure called for in the contract and identified in the exchange file.
- **Enabling Automated Repository Check**-In - as a STEP model is checked into a PDM system, an automated validation routine checks the model using the schema (located in the DD service) that is identified in the Part 21 data file
- **Improving Quality of Data Handoffs** - an MCAD geometry model is sent from design to thermal analysis and validation is performed using the correct schema version as referenced in the model; validation is an automated process that occurs before any work is done with the model as it is transferred between domains
- **Validating for Adequacy and Range** – the PDS (NASA’s Planetary Data System) central node receives a dataset description in template format to be ingested into the dataset catalogue database. Automatically, a standard validation routine is performed using the data

dictionary service that checks for required keywords, key word values and value types in the dataset in template format against a corresponding data structure stored in the PDS domain of the data dictionary service.

Data Modeling Scenarios

- **Data Reuse in Modelling Activities**- a data modeller, charged with developing an information model for a new application, uses data elements published in several DD services (much like a parts library). Reusing existing data elements and including references in the newly created data model ensures that the related datasets will have compatible interfaces and will be able to share and integration information across domains.
- **Creating a TDP (technical data package)** - an application performs a schema check against objects about to be wrapped into a TDP (e.g., STEP AP232 or PDM Schema TDP) to ensure their correct structure and meta-data content to be used in populating TDP data structures. This will allow those later using the TDP to understand the nature of the components contained in the TDP.
- **Data Integration Enabled** - an analyst, charged with integrating data from two or more data sets, accesses the “correct” version of each schema as referenced in the data set from the “DD service space” allowing them to identify/map interfaces between the data sets, e.g., MCAD-ECAD-fabrication-cost data
- **Extending a schema** - to solve a "local" problem, a data modeller searches for data elements in the data dictionary service that could be used to solve the local problem. The search returns a list of data elements from a published collection of data items to extend an existing “official” schema; the new schema is published in the DD service with traces/links back to the “official” schema

Conclusions

Today we find ourselves living in a world that is becoming increasingly interconnected but strangely we find most datasets isolated and disconnected from each other. It is important to build the architecture and mechanisms that provide connectivity between datasets. A data dictionary service that provides common access to the semantics and data structures is essential if information is to be understood, reused and integrated.

Combining the LDAP protocol with the ISO 11179 data element set leverages and combines two established standards to create a new capability. This approach follows an age old technique of combining two existing technologies to create a new third technology. This approach builds on top of and provides connection to the evolutionary environment upon which the Internet born. The purpose of a standards-based Data Dictionary Service is not to create uniformity for uniformity sake but to enhance communication because only through effective communication can we determine what we have in common and where features and content are unique.

References

Berners-Lee, T., *Weaving the Web*, Harper Information, May 2000, ISBN 0-06-251587-X

Comer, Douglas, *Internetworking with TCP/IP, Prentice Hall*, 1998, ISBN 0-13-470154-2

Hughes, J et al, *The Management of Meta-data within the Planetary Data System*, a JPL whitepaer, http://www.llnl.gov/liv_comp/metadata/papers/pos-papers-may-1994/StevenHughes

IBM Dictionary of Computing [10th Edition], McGraw-Hill, 1994

ISO/IEC 11179-1, Specification and Standardization of Data Elements, 1999

ISO 10303-1, Industrial Automation Systems and Integration – Product Data Representation and Exchange, 1994

IETF, *Internet RFC-1310, Internet Standards Process,* http://www.ietf.org/rfc/rfc1310.txt?number=1310

IETF, *RFC-1777, Lightweight Directory Access Protocol,* http://www.ietf.org/rfc/rfc1310.txt?number=1777

IETF, *Internet RFC-1778 ,The String Representation of Standard Attribute Syntaxes,* http://www.ietf.org/rfc/rfc1310.txt?number=1778

IETF, *Internet RFC-1779, A String Representation of Distinguished Names,* http://www.ietf.org/rfc/rfc1310.txt?number=1778

IETF, *Internet RFC-1959, An LDAP URL Format,* http://www.ietf.org/rfc/rfc1310.txt?number=1959

IETF, *Internet RFC-1823, The LDAP API,* http://www.ietf.org/rfc/rfc1310.txt?number=1823

Issak, J. et al, *Open System Handbook*, IEEE Standards Press, 1994, ISBN 1-55937-435-7

Howes, T. et al, *Understanding and Deploying LDAP Directory Services*, Macmillan Publishing, December 1998, ISBN: 1-57-870070-1

Markey, B., *A System Administrator's View of LDAP*, Netscape, 1998, http://people.netscape.com/bjm/whyLDAP.html

Newton, J. , Application of Metadata Standards, NIST, 1996, http://www.computer.org/conferences/meta96/newton/paper.html

Wilcox, B., *Implementing LDAP*, Wrox Press, Inc., March 1999, ISBN 1-86100-221-1

Appendix A - Glossary of Related Terms

attribute - A characteristic of an object or entity ISO11179

base DN – the distinguished name that identifies the starting point of a search. e.g. to search for all entries relating to people at XYZ Inc, the base DN for such a search would be “ou=people, o=XYZ.com”

data element - A unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes ISO11179

DN – [Distinguished Name] – an term used to uniquely identify each entry in an LDAP directory and is made up of attribute:value pairs, separated by commas.

DTD – [Document Type Definition] - a language that describes the contents of an SGML or XML document; provides definitions that may be embedded within an XML document or in a separate file; DTDs are expected to be replaced by W3C XML schemas. W3C

IEC – [International Electrotechnical Commission] - a standards body that sets international electrical and electronics standards; it is made up of national committees from over 40 countries.

ISO – [International Standards Organization] – an organization that sets international standards, founded in 1946. The U.S. member body is ANSI. ISO deals with all fields except electrical and electronics, which is governed by the older International Electrotechnical Commission (IEC). With regard to information processing, ISO and IEC created JTC1, the Joint Technical Committee for information technology.

IETF – [Internet Engineering Task Force] – an organization of working groups dedicated to identifying problems and proposing technical solutions for the Internet.

LDAP – [Lightweight Directory Access Protocol] an Internet standard that provides mechanisms and structures for access on-line directory information.

namespace – (1) a name or group of names that are defined according to some naming convention; a flat namespace uses a single, unique name for every device where a hierarchical namespace partitions the names into categories known as top level domains. e.g. the Internet uses a hierarchical namespace that partitions the names into top level domains such as .com, .edu and .gov, etc., which are at the top of the hierarchy. TechWeb Glossary ; (2) van XML namespace is a collection of names, identified by a URI reference [RFC2396], which are used in XML documents as element types and attribute names. XML namespaces differ from the "namespaces" conventionally used in computing disciplines in that the XML version has internal structure and is not, mathematically speaking, a set. W3C ; (3) a collection of vocabulary, data elements and data structures associated with an application domain. JAU

referral – refers an LDAP client to another LDAP server. An LDAP server can be configured to send your client a referral if your client requests a DN with a suffix that is not in the server’s directory tree (e.g. if the directory includes entries und `“o=Airius.com”` and your client requests an entry under `“o=AiriusWest.com`”). Referrals contain LDAP URLs that specify the host, port and base DN of another LDAP server.Netscape LDAP glossary

RFC – [Request for Comment] – an Internet convention used to document Internet standards; a document that describes the specifications for a recommended technology. RFCs are used by the Internet Engineering Task Force (IETF) and other standards bodies. First used during the creation of the ARPAnet protocols in the 1970s, the IETF has published more than 2500 RFCs, all of which can be viewed at www.ietf.org/rfc.html. TechWeb Glossary

schema – (1) those definitions which describe the concept of the data and the relationship between the various elements or components of the data. A collection of items forming part or all of a model ISO 10303-11: 1994.; (2) schema can be viewed as a collection (vocabulary) of type definitions and element declarations whose names belong to a particular namespace called a target namespace that enable us to distinguish between definitions and declarations from different vocabularies.W3C a definition of data structure Dacom, 1985

search reference – [a.k.a. continuation reference or smart referrals] an entry in the directory that refers to another LDAP server; search references are returned in search results along with entries foun in the search as opposed to a referral which is returned if the base DN does not have a suffix that is handled by the server. Netscape LDAP glossary

semantics - the branch of linguistic science which deals with the meaning of words Webster

wrapper - a data structure or software that contains ("wraps around") other data or software, so that the contained elements can exist in the newer system. TechWeb Glossary

Appendix B - Configuration of the prototype Distributed Data Dictionary Service

The first node of the prototype **D^3** Service (Distributed Data Dictionary Service) was setup at the Jet Propulsion Laboratory at step.jpl.nasa.gov. The prototype can be viewed at:

`http://step.jpl.nasa.gov/ldap`

To evaluate the feasibility and applicability, the prototype is being developed with a minimalist approach using surplused equipment and open source software.

Hardware: (a ~ 3 year old PC workstation salvaged from the retirement list)

- 166 mhz Intel Pentium
- 64 mb ram
- 2 GB hard drive

Operating System:

- Linux (Mandrake 7.0)   [also tested on RedHat 6.2 and SuSE 7.1]

Software and documentation available commercially or on-line at
http://www.mandrake.com/

LDAP Server Software:

- OpenLDAP slapd 1.2.6-Release

Open Source software. Binaries and documentation downloaded from

http://www.OpenLDAP.org/

The LDAP service is control through several key files.

The `slapd.conf` file looks like:

```
   # See slapd.conf(5) for details on configuration options.
   # This file should NOT be world readable.
   #
   include         /usr/local/etc/openldap/slapd.at.conf
   include         /usr/local/etc/openldap/slapd.oc.conf
   schemacheck     off
   #referral       ldap://ldap.itd.umich.edu

   pidfile         /usr/local/var/slapd.pid
   argsfile        /usr/local/var/slapd.args

   #######################################################################
   # ldbm database definitions
   #######################################################################

   database        ldbm
   suffix  "dc=JPL, dc=US"
   directory       /usr/tmp
   rootdn          "cn=root, dc=JPL, ,dc=NASA, dc=gov"
```

```
rootpw          ******
# cleartext passwords, especially for the rootdn, should
# be avoided.  See slapd.conf(5) for details.
```

Information was loaded into the prototype using LDIF (Lightweight Directory Interchange Format) files..A sample LDIF input looks like:

```
dn: cn=description, dc=JPL, dc=US
cn: description
o: Jet Propulsion Laboratory
objectclass: organization
objectclass: dcObject
```

Included with most LDAP servers is a command line utility called `ldapadd` that moves correctly formatted information in LDIF files into an LDAP database. Here is a sample `ldapadd` command line exchange with the verbose (-v) feedback switched on:

```
# ldapadd -v -D "cn=root, dc=JPL, dc=US" -W < test4.ldif
Enter LDAP Password:
add cn:
        description
add o:
        Jet Propulsion Laboratory
add objectclass:
        organization
        dcObject
adding new entry cn=description, dc=JPL, dc=US
modify complete
```